

Universidad Autónoma de Nuevo León  
Facultad de Ingeniería Mecánica y Eléctrica

Biomecánica

Práctica #1

Maestro: Isaac Estrada

Matrícula	Nombre	Carrera
1908186	Armando Rincón Reyes	IMTC
1991920	Idalia Rivera Del Angel	IMTC
1910192	Alan Magdiel Villa Herrera	IMTC
1910912	Mónica Rangel Guerra	IMTC
1900416	Diego Alexis Limón Báez	IMTC
1927564	Marcos Fernando Romero Carrillo	ITMC

Hora: N3

Grupo: 202

Semestre: 7mo

Fecha: 5 de septiembre 2022

Semestre Agosto-Diciembre 2022

Ciudad Universitaria, San Nicolás de los Garza.

## 1. Objetivo

El estudiante conocerá cada una de las secciones que integran el código de optimización topológica, como se debe de crea el archivo (.m) en MATLAB y como se ejecuta el análisis.

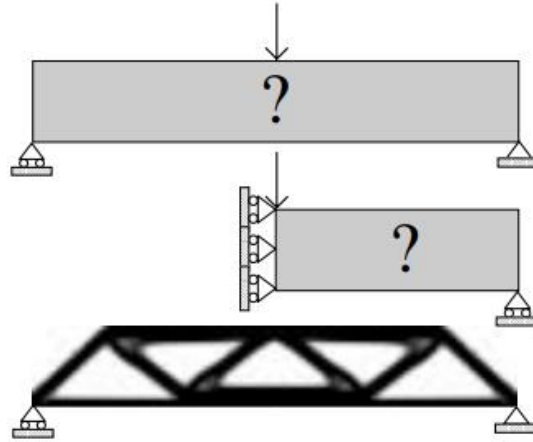
## 2. Nombre y definición de la programación mencionar un ejemplo de forma de la GEOMETRÍA

La optimización de topología hace referencia a un proceso el cual tiene como objetivo el ver qué elementos son los necesarios para que un objeto, estructura o material siga cumpliendo con los requerimientos de esfuerzo y resistividad eliminando aquellos elementos que agreguen peso, masa o material al objeto a optimizar. Este método es puramente matemático pues hace uso de diferentes matrices para calcular las posiciones en las cuales es posible optimizar el objeto. Según la página de Hisour Arte Cultura Historia define la optimización de topología (TO) como: “un método que optimiza el diseño del material dentro de un espacio de diseño dado, para un conjunto dado de cargas, condiciones de contorno y restricciones con el objetivo de maximizar el rendimiento del sistema.”

La optimización de topología tiene una amplia gama de aplicaciones en ingeniería aeroespacial, mecánica, bioquímica y civil. Actualmente, los ingenieros usan principalmente TO en el nivel de concepto de un proceso de diseño. Debido a las formas libres que ocurren naturalmente, el resultado es a menudo difícil de fabricar.

El código desarrollado por O. Sigmund tiene como objetivo minimizar el cumplimiento de cargas estáticas en estructuras. El código de optimización topológica de 99 líneas en Matlab que se utilizará en este laboratorio se divide en 36 líneas para la programación principal, 12 líneas para los criterios de optimización, 16 líneas para el filtro de mallado y 35 líneas para el código del elemento finito. Un algoritmo de optimización entonces busca, mediante iteración, resolver las densidades de los elementos los cuales minimizan el cumplimiento global de la estructura.

Se realizan varias simplificaciones para el código de Matlab. El dominio del diseño se asume como rectangular y discretizado por elementos cuadrados infinitos. De esta manera el número de elementos y de nodos es más simple en el análisis de la viga. De manera que se pueda asegurar una solución a la optimización de topología algún tipo de restricción tienen que ser introducidas. Una de las técnicas usadas en el trabajo de Sigmund es la de técnica de filtrado asegurando entonces que produzca una independencia de mallado. En la siguiente imagen se muestra la figura a analizar, solo se analizará una sección de la viga.



El método de filtro de independencia de mallado consiste en cambiar los elementos de la sensibilidad como se muestra en la ecuación:

$$\widehat{\frac{\partial c}{\partial x_e}} = \frac{1}{x_e \sum_{f=1}^N \hat{H}_f} \sum_{f=1}^N \hat{H}_f x_f \frac{\partial c}{\partial x_f}.$$

Es importante recalcar que este método viene a reemplazar el antigua labor de iteración de los ingenieros para poder mejorar el diseño de un material, con TO solo es necesario introducir los datos necesarios para que el software o líneas de código puedan arrojar resultados más rápidos permitiendo así el ahorro de dinero además de salvar tiempo.

El objetivo principal de la optimización es la de maximizar la rigidez del material en general o minimizar su cumplimiento en una cantidad determinada de eliminación de masa. El cumplimiento se define como una medida de flexibilidad y es el recíproco de la rigidez. El cumplimiento global es la suma de elementos elásticos o las energías de deformación como se muestra en la ecuación:

$$\min C(\{\rho\}) = \sum_{e=1}^N (\rho_e)^p [u_e]^T [K_e] [u_e]$$

Donde:  $(u_e)$  es el vector de desplazamiento nodal del elemento  $e$ ,  $(K_e)$  es la rigidez del elemento  $e$ , y el vector  $(\rho)$  contiene las densidades relativas de los elementos  $p_e$ .

### 3. Estado del arte

La optimización topológica es una técnica que pertenece al análisis estructural y consiste básicamente en analizar un componente o estructura y dependiendo de cómo se cargó, eliminar el material no deseado.

La formulación convencional de la optimización topológica (TO) utiliza un método de elementos finitos para evaluar el desempeño del diseño. El diseño se optimiza utilizando técnicas de programación matemática basadas en gradientes, como el algoritmo de criterios de optimalidad y el método de mover asíntotas o algoritmos no basados en gradientes, como los algoritmos genéticos.

La optimización de topología tiene una amplia gama de aplicaciones en ingeniería aeroespacial, mecánica, bioquímica y civil. Actualmente, los ingenieros usan principalmente TO en el nivel de concepto de un proceso de diseño. Debido a las formas libres que ocurren naturalmente, el resultado es a menudo difícil de fabricar. Por esa razón, el resultado que surge de TO a menudo se ajusta para la fabricación. Agregar restricciones a la formulación para aumentar la capacidad de fabricación es un campo activo de investigación. En algunos casos, los resultados de TO se pueden fabricar directamente utilizando fabricación aditiva; Por lo tanto, TO es una parte clave del diseño para la fabricación aditiva.

En el proceso de optimización topológica, se deben tener en cuenta varios aspectos; el espacio de diseño, el o los casos de carga que va a sufrir la pieza en cuestión, el material y la tecnología con que se va a realizar su fabricación, la reducción de costes mediante la minimización de soportes y aprovechamiento de la cuba de impresión, en caso de utilizar tecnologías aditivas, y muchos más.

El software de optimización topológica cada vez más sofisticado permite a los ingenieros guardar el material posible para un objeto mientras mantienen o mejoran su fuerza o flexibilidad (si es necesario) y teniendo en cuenta las restricciones que se colocarán en él. Anteriormente, se basa en la intuición, el método de prueba y error y / o el genio de los creadores y / o ingenieros de fabricación.

MATLAB, el lenguaje de cálculo técnico desarrollado por MathWorks, es un entorno de programación para el desarrollo de algoritmos, análisis de datos, visualización y cálculo numérico.

Los usos más comunes de MATLAB incluyen cálculos matriciales, desarrollo y ejecución de algoritmos, creación de interfaces de usuario y visualización de datos. El entorno informático numérico de paradigmas múltiples permite a los desarrolladores interactuar con programas desarrollados en diferentes lenguajes, lo que permite aprovechar las fortalezas únicas de cada idioma para diversos fines.

El código de optimización topológica en Matlab que se utilizara en este laboratorio se divide en 36 líneas para la programación principal, 12 líneas para los criterios de optimización, 16 líneas para el filtro de mallado y 35 líneas para el código de elemento finito.



## 4. Procedimiento de la programación

Para comenzar nuestra programación de optimización procederemos a abrir el software MATLAB y abrimos un New Scrip y guardamos el archivo para su próxima ejecución (Matlab pide guardar el documento para después ejecutarlo).

```
Editor - C:\Users\marco\Documents\7mo semestre\Practica1\Practica1.m
Practica1.m x +
1 %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%%
2 %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%%
```

Una vez guardado el nombre del archivo procedemos a colocar el código, verificamos los errores que tenga y corregimos para después ejecutar.

```
1 %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%%
2 %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%%
3 %Equipo7 Lab Biomecanica N3%
4 function Practical(nelx,nely,volfrac,penal,rmin);
5 % INITIALIZE
6 x(1:nely,1:nelx) = volfrac;
7 loop = 0;
8 change = 1.;
9 % START ITERATION
10 while change > 0.01
11     loop = loop + 1;
12     xold = x;
13     % FE-ANALYSIS
14     [U]=FE(nelx,nely,x,penal);
15     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
16     [KE] = lk;
17     c = 0.;
18     for ely = 1:nely
19         for elx = 1:nelx
20             n1 = (nely+1)*(elx-1)+ely;
21             n2 = (nely+1)* elx +ely;
22             Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
23             c = c + x(ely,elx)^penal*Ue'*KE*Ue;
24             dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
25         end
26     end
```



# UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



# FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

```

Practica1.m
27 % FILTERING OF SENSITIVITIES
28 [dc] = check(nelx,nely,rmin,x,dc);
29 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
30 [x] = OC(nelx,nely,x,volfrac,dc);
31 % PRINT RESULTS
32 change = max(max(abs(x-xold)));
33 disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
34       ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
35       ' ch.: ' sprintf('%6.3f',change) ])
36 % PLOT DENSITIES
37 colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
38 end
39 %***** OPTIMALITY CRITERIA UPDATE *****
40 function [xnew]=OC(nelx,nely,x,volfrac,dc)
41 l1 = 0; l2 = 100000; move = 0.2;
42 while (l2-l1 > 1e-4)
43     lmid = 0.5*(l2+l1);
44     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
45     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
46         l1 = lmid;
47     else
48         l2 = lmid;
49     end
50 end

```

```

Practica1.m
51 %***** MESH-INDEPENDENCY FILTER *****
52 function [dcn]=check(nelx,nely,rmin,x,dc)
53 dcn=zeros(nely,nelx);
54 for i = 1:nelx
55     for j = 1:nely
56         sum=0.0;
57         for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
58             for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
59                 fac = rmin-sqrt((i-k)^2+(j-l)^2);
60                 sum = sum+max(0,fac);
61                 dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
62             end
63         end
64         dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
65     end
66 end
67 %***** END *****

```

```

67  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68  function [U]=FE(nelx,nely,x,penal)
69  - [KE] = lk;
70  - K = sparse(2*(nelx+1),(nely+1), 2*(nelx+1)*(nely+1));
71  - F = sparse(2*(nely+1),(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
72  - for elx = 1:nelx
73  -   for ely = 1:nely
74  -     n1 = (nely+1)*(elx-1)+ely;
75  -     n2 = (nely+1)* elx +ely;
76  -     edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
77  -     K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
78  -   end
79  - end
80  % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
81  - F(2,1) = -1;
82  - fixeddofs = union([1:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
83  - alldofs = [1:2*(nely+1)*(nelx+1)];
84  - freedofs = setdiff(alldofs,fixeddofs);
85  % SOLVING
86  - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
87  - U(fixeddofs,:)= 0;

88  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
89  function [KE]=lk
90  - E = 1.;
91  - nu = 0.3;
92  - k=[ 1/2-nu/6    1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
93  -    -1/4+nu/12 -1/8-nu/8    nu/6      1/8-3*nu/8];
94  - KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
95  -                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
96  -                  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
97  -                  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
98  -                  k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
99  -                  k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
100 -                  k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
101 -                  k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

De las líneas 1 a la 36 se comienza a distribuir el material uniformemente en el dominio del diseño (línea 4), posteriormente el bucle principal comienza con la subrutina de elementos finitos (línea 14) que devuelve el vector de desplazamiento U. La subrutina de rigidez está en la línea 14 y después de la 18 a la 25 es un bucle de todos los elementos.

El análisis de sensibilidad va seguido de una llamada al filtro de independencia de malla (línea 29) y el optimizador de criterios de optimalidad (línea 30). La conformidad actual, así como otros parámetros, se imprimen por las líneas 31-35 y se representa la distribución de densidad resultante (línea 36). El bucle principal se termina si el cambio en las variables de diseño (el cambio determinado en la línea 31) es inferior al 1 por ciento. Sigue el Optimizador basado en criterios de optimalidad (líneas 37-49). Las variables de diseño actualizadas son encontradas por el optimizador. Sabiendo que el volumen de material (sum(sum(xnew))) es una función monótonamente decreciente del multiplicador de Lagrange (lag), el valor del multiplicador de Lagrangian que satisface la restricción de volumen se puede



encontrar por un algoritmo de dos secciones (líneas 41-48). El algoritmo de bifurcación se inicializa adivinando un  $I_1$  inferior y un  $I_2$  superior ligado al multiplicador lagrangiano (línea 40). El intervalo que limita el multiplicador lagrangiano se reduce repetidamente a la mitad hasta que su tamaño es menor que los criterios de convergencia.

Sigue el Filtrado de la independencia de malla (líneas 50-65) representan la implementación de Matlab. Por último, el Código del elemento finito (líneas 66-100). La matriz de rigidez global está formada por un bucle sobre todos los elementos (líneas 71-78). Las variables  $n_1$  y  $n_2$  denotan números de nodos de elementos superior izquierdo y derecho en números de nodos globales y se utilizan para insertar la matriz de rigidez de elementos en los lugares correctos en la matriz de rigidez global. La matriz de rigidez del elemento se calcula en las líneas 87-100. La matriz de 8 por 8 para un elemento cuadrada bi-lineal de 4 nodos se determinó analíticamente utilizando un software de manipulación simbólica. El módulo  $E$  de Young y la relación  $\nu$  de Poisson se pueden alterar en las líneas 89 y 90.

## 5. Implementación o desarrollo de la programación en sus diferentes vistas

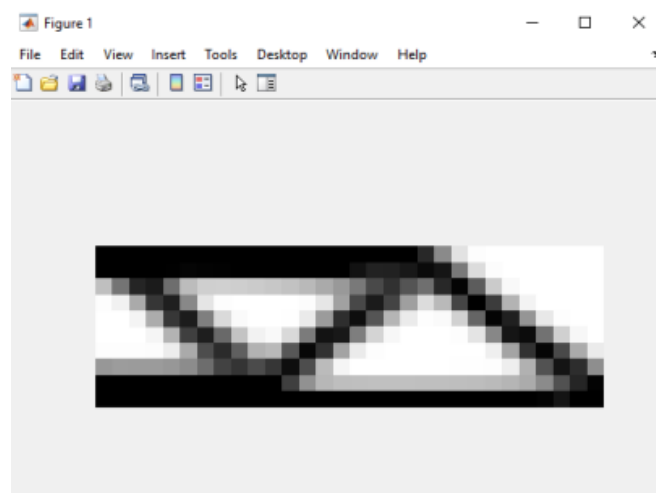
Teniendo en cuenta las siguientes funciones de nuestro código inicial:

```
function Practical(nelx,nely,volfrac,penal,rmin);
```

Donde:

- nelx y nely: son el número de elementos en las direcciones horizontal y vertical.
- volfrac: es la fracción de volumen
- penal: es el poder de penalización
- rmin: es el tamaño del filtro (dividido por el tamaño del elemento).

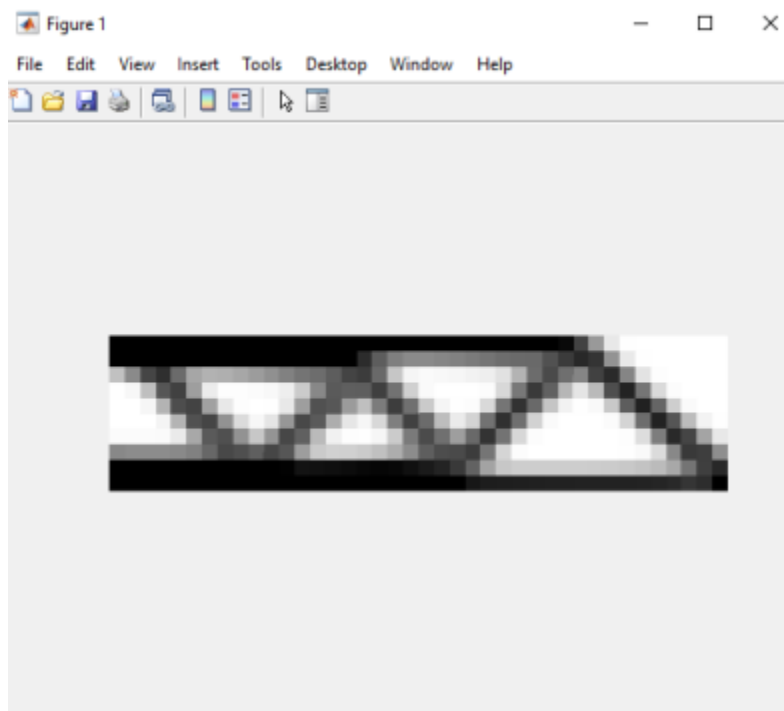
Dando los parámetros (30,10,0.5,3.0,1.5) obtendremos:



*Ilustración 1. Viga Optimizada*

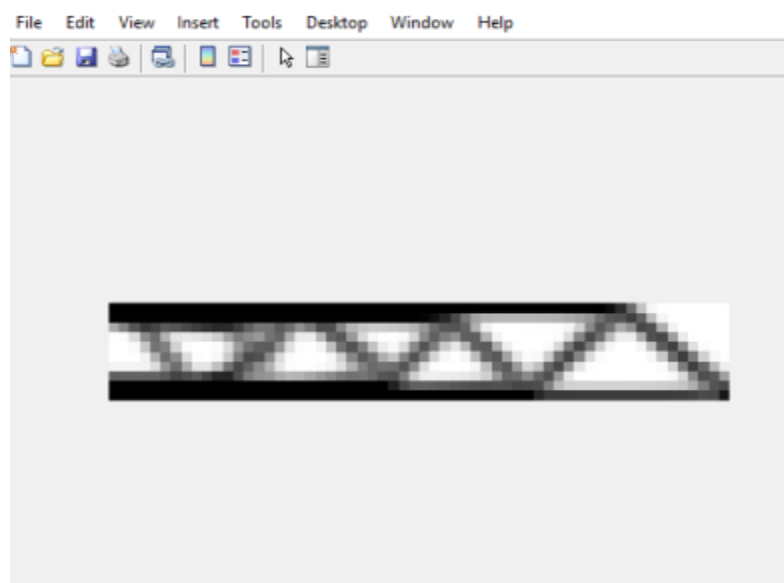


Segundo parámetro: (40,10,0.5,3.0,1.5) :



*Ilustración 2. Segunda Viga*

Tercer parámetro: (60,10,0.5,3.0,1.5)



*Ilustración 3. Tercer Viga*

## 6. Conclusiones por cada autor

### **Alan Magdiel Villa Herrera**

Para esta primera práctica de laboratorio se implementó un código de optimización topológica en el software de Matlab, la TO ha demostrado ser una gran herramienta en el diseño de elementos, lo cual es de suma relevancia, ya que en la biomecánica es bien conocido que es necesario encontrar el diseño de piezas óptima para el cuerpo humano en prótesis.

Personalmente había utilizado Matlab para otro tipo de aplicaciones y al conocer que se puede usar en técnicas de optimización topológica donde se elimina material no deseado y se puede generar un análisis de elemento finito, me hace pensar en el nivel de complejidad y aplicación que podemos llevar un software que ya hemos estudiado con anterioridad.

Es una problemática común dentro de la ingeniería determinar la mejor configuración geométrica de una pieza que proporcione cierta función dependiendo de la aplicación que tenga dicha pieza, por lo que la optimización topológica es de las mejores estrategias que podemos encontrar para solucionar este problema.

Una de las principales dificultades con la practica fue precisamente que no se había tratado Matlab con una aplicación parecida, pero con la organización del equipo y con el uso de fuentes se pudo profundizar, analizar y aplicar mejor el tema de estudio.

### **Mónica Rangel Guerra**

Personalmente yo no llevé la materia de elemento finito, sin embargo, esta es una técnica utilizada en esta materia. La técnica analizada en esta actividad nos es de gran ayuda ya que nos ayudan a que las estructuras sean más ligeras, ya que no crean huecos en su interior. La técnica de optimización topológica puede ser aplicada en diversas áreas, una de ellas es la biomédica para el desarrollo de prótesis.

### **Marcos Fernando Romero Carrillo**

Puedo decir que la optimización tiene un amplio uso en la ingeniería ya que nos sirve para poder reducir costos, mejorar propiedades o incluso mejorar el tiempo de vida de algún producto o diseño. En esta práctica vimos algo relacionado y fue respecto a la programación de vigas y del cómo podríamos optimizar, el código no fue tan difícil de comprender solo fue cuestión de hacer modificaciones y observar el cómo las figuras cambiaban respecto a los nuevos parámetros dados. Podría decir que a futuro esto puede ser usado para el diseño y la fabricación de prótesis o incluso puede ser que ya está siendo usado para este campo de ingeniería

### **Diego Alexis Limón Báez**

Durante esta práctica se pudieron entender los conceptos básicos de optimización de topografía y como esta es ampliamente en varias áreas de la ingeniería e incluso de la medicina moderna. Prácticamente consta de eliminar aquella masa innecesaria para que un producto sea más barato sin perder sus propiedades. Este concepto además de un ejemplo de su programación nos da como ingenierías varias herramientas con las cuales podemos hacer diseños más optimizados de tal forma que podemos manufacturar dichas piezas de la mejor manera. Vimos el ejemplo de MATLAB, pero también puede ser aplicado en diferentes software y todos ellos con la misma finalidad. En este ejemplo se eligió un método en específico de optimizado, pero en realidad existen muchas otras formas y ya dependerá de nuestro criterio saber cuál elegir.

### **Idalia Rivera Del Ángel**

En la realización de la práctica se obtuvieron conocimientos acerca de cómo la optimización topológica nos es útil para obtener la distribución óptima de un material en un dominio predefinido para que cumpla con las condiciones dadas. La utilización de este método nos permite llevar a cabo el diseño del material dentro del espacio dado, para así maximizar el rendimiento del sistema. Mediante la implementación del código de optimización topológica, en esta práctica buscamos minimizar el cumplimiento de estructuras cargadas estáticamente.

### **Armando Rincón Reyes**

El desarrollo de esta práctica me ayudó a comprender nuevos conceptos de topología que no conocía, así como recordar comandos y sintaxis que se utilizan en Matlab, ya que es algo que no usaba desde hace algunos semestres. Aprendí personalmente que esto de la optimización es un proceso con varias etapas y algo largo, que es difícil de llevar a cabo sino se tienen los conocimientos adecuados o se tiene tanta experiencia con ciertos programas como en este caso Matlab.

## 6. Referencias bibliográficas

- Hisour. (2019). hisour. Obtenido de <https://www.hisour.com/es/topology-optimization-40760/>
- Reclu IT. (23 de Julio de 2020). Reclu IT. Obtenido de <https://recluit.com/que-es-matlab/#.Yxc3tXbMKUk>
- CATEC. (2019). Optimización Topológica. septiembre 5, 2022, de CATEC Sitio web: <http://www.catec.aero/es/materiales-y-procesos/l%C3%ADnea-de-investigaci%C3%B3n/optimizaci%C3%B3n-topol%C3%B3gica>
- Anónimo. (s.f.). Método SIMP para optimización de topología. septiembre 5, 2022, de DASSAULT SYSTEMS Sitio web: [https://help.solidworks.com/2019/spanish/SolidWorks/cworks/c\\_simp\\_method\\_topology.htm](https://help.solidworks.com/2019/spanish/SolidWorks/cworks/c_simp_method_topology.htm)
- Lucía C.. (2020). La optimización topológica en la impresión 3D. septiembre 5, 2022, de 3D natives Sitio web: <https://www.3dnatives.com/es/optimizacion-topologica-10012017/#!>